

EMITTER International Journal of Engineering Technology

ISSN: 2443-1168, Vol. 8, No. 2, December 2020, pp. 459~476

DOI : 10.24003/emitter.v8i2.566

Performance Comparison of Genetic and Greedy Algorithms in Underlay Device-to-Device Communication

Salma Pratiwi¹, Arfianto Fahmi², Vinsensius Sigit Widhi Prabowo³School of Electrical Engineering
Telkom University, BandungE-mail: salmaprtw@student.telkomuniversity.ac.id¹,
arfiantof@telkomuniversity.ac.id², vinsensiusvsw@telkomuniversity.ac.id³*Received September 14, 2020; Revised October 16, 2020; Accepted December 9, 2020*

Abstract

The number of mobile devices continues to grow across the world and impacts the overall network load. An increasing number of devices that connected to the main network will harm the system's quality of services. To solve this problem, LTE-A technology introduce the Device-to-Device (D2D) Communication. D2D enables direct communications between users without routing via eNB, so the network load could be maintained. D2D scheme also can be underlayed on an existing cell using same resource with the conventional cellular users (CU). A good resource allocation algorithm is needed to achieve a proper quality of services and to maintain the interference level between D2D users and CUs. This works tries to analyze the performances of Genetics based resource allocation algorithm and then compared it with the conventional greedy algorithm. In general, Genetic algorithm has three steps to allocates resource block (RB) to each user, proportional selection, crossover, and mutation, and these three steps executed repeatedly to achieve the best performances. From the simulation result, Genetic algorithm can achieve better energy efficiency, lower usage power, and lower interference level. Although genetic algorithm caused lower system's sumrate and spectral efficiency.

Keywords: Device-to-Device (D2D), Genetic Algorithm, Spectral Efficiency, Energy Efficiency, Interference Mitigation.

1. INTRODUCTION

The number of mobile communication users continues to increase all over the world. This raises the problem of increasingly crowded communication traffic in cellular networks. One solution to this problem is the Device-to-Device (D2D) communication technology. LTE enables the usage of heterogeneous network in the system such as Femtocells and D2D communications [1]. D2D communication as a technology component for LTE-A enables a direct connection between devices, without rerouting the traffic through any network infrastructure. Under normal conditions, the network

controls the usage of direct links radio resource minimize the level of interference. This technology promises to increase coverage, offload backhaul, provide fallback connectivity, and increase spectrum utilization and capacity per area [2]. D2D communication has two types based on the used spectrum by D2D users and their impact on cellular users, which are inband and outband D2D communication [3].

Outband D2D communications exploit the unlicensed spectrum. The advantages of this scheme are the interference between D2D links and cellular links is minimum, since D2D communications use license-exempt bands, similar to Wi-Fi Direct, Wi-Fi, Bluetooth, and Ultra-Wideband technologies.

In the inband scheme, D2D communications communicate through a licensed spectrum, which is also being used by conventional cellular users. There are two types of inband scheme, underlay D2D communications and overlay D2D communications. Overlay scheme allocates dedicated links for D2D users, meanwhile in underlay scheme, D2D user utilize the same spectrum with some cellular users.

Overlay scheme offers a good system performance, since D2D users can communicate without co-channel interference on dedicated resources, but will have a low spectral efficiency. On the other hand, underlay scheme share the same spectrum resources with some other cellular user equipments (CUEs). With spectrum sharing, D2D communications can improve spectrum efficiency and network throughput which are two important performance indices. Nevertheless, mutual interference between D2D user equipments (DUEs) and CUEs could critically affect the system performance [3]. Thus, a good resource allocation algorithm to allocate radio resources properly is needed to maintain the system performances.

2. RELATED WORKS

There are several research related to resource allocation in underlay D2D communication [4-7]. In [4], the research analyzes system performance based on heuristic algorithm, then compares it with minimum interference algorithm and random allocation algorithm. The study presented a simulation result of a heuristic algorithm working optimally when the system has a dense number of D2D pairs.

Work [5] proposed a joint mode selection allocation algorithm. The research developed a complex branch-and-bound method to have low complexity for D2D mode selection and channel allocations problems. This works proposed two steps allocation process, which are power control and channel determination that executed independently.

Work [6] proposed resources allocation based on Genetic algorithms to minimize the interference level among users, maximize the spectral efficiency, and maintain the value of Signal to Interference plus Noise Ratio (SINR) of cellular users due to D2D communication. That study used the channel prediction method and evolves toward global maximum by searching different parts of search space simultaneously.

In [7], process allocation of resource in uplink way with D2D pair used Join Greedy algorithm, then compares it with the conventional greedy algorithm. Based on the simulation results, the joint greedy allocation scheme achieve better efficiency and fairness compared with greedy allocation algorithm.

This works analyze and compare the performance parameters between Genetic and greedy algorithm on D2D communication systems, including system interference, system sum-rate, system spectral efficiency, total energy system, and system energy efficiency. The rest of paper is composed as follows. In Section 3, the concepts those can solve the problem stated in the Introduction is explained. The system model is described in Section 4. Numerical simulation results are presented in Section 5. Finally, the conclusions of the paper are given in Section 6.

3. ORIGINALITY

This works tries to modify the genetic based algorithm in [6] and compared with the greedy algorithm. The modification lies in the structure of the two main process of algorithm, which are the mutation and the crossover. The genetic based algorithm has no restriction on a number of D2D pairs that can use same RB with cellular user in an instance, but a minimum value of SINR is required. Genetic based algorithm is being used as a comparison. This works tries to evaluate the genetics algorithm performances in several parameters, such as spectral efficiency, energy efficiency, total power usage, interference level, and the system's sumrate. This works done to give a better explanation about the performances of genetics allocation algorithm.

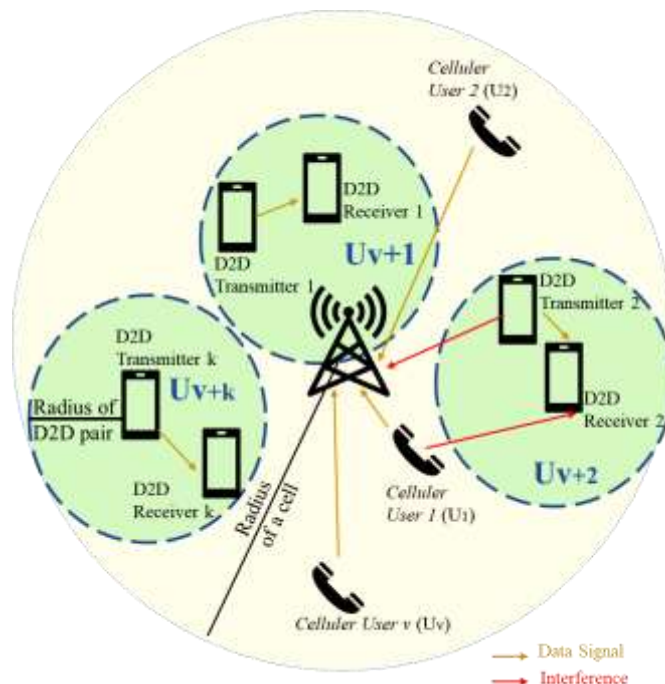


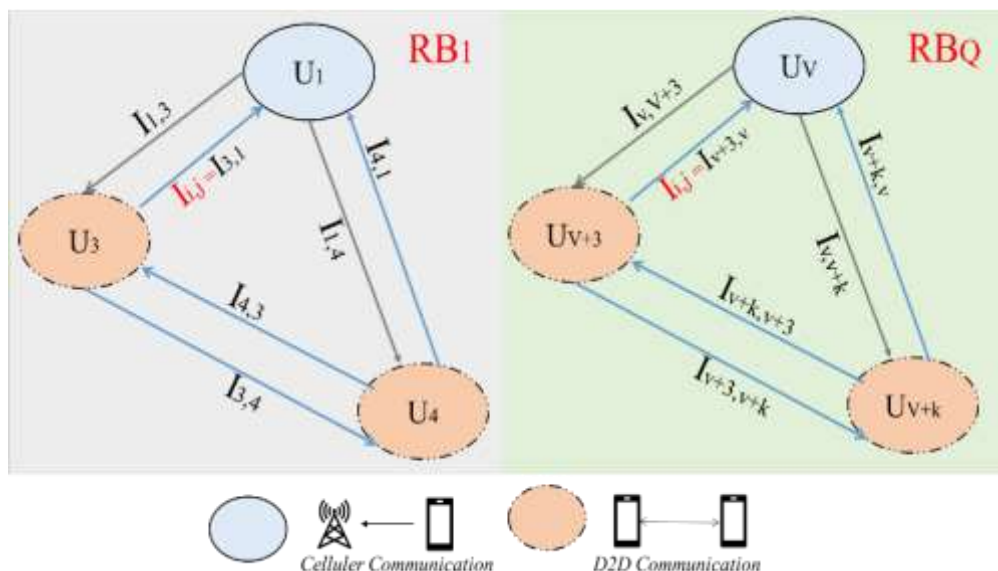
Figure 1. The model systemAn example of an interference graph in D2D communication

4. SYSTEM DESIGN

System model that being used in this research is a single cell with one eNB in the center. There are two type of user, which are cellular users (CUEs) that communicates through eNB, and D2D users (DUEs) which can communicates directly between device and form a DUE pairs. These users are randomly distributed on cell area. The cell model can be seen on Figure 1

The pair of communications were defined as $N = \{U_1, U_2, \dots, U_V, U_{V+1}, \dots, U_{V+K}\}$, where U_i showed the i^{th} communication, then V and K is the total number of CUEs and DUE pairs respectively. The set of CUEs is noted by $M = \{U_1, U_2, \dots, U_V\}$, and the set of D2D pairs was defined by $D = \{U_{V+1}, U_{V+2}, \dots, U_{V+K}\}$. The set of resources block (RB) denoted by $R = \{RB_1, RB_2, \dots, RB_Q\}$. Where Q denotes the total number of RBs. This model system used a matrix with size $Q \times (V+K)$ with an element of $a_{r,i}$. an element of $a_{r,i} = 1$ if i^{th} user utilizes the r^{th} of RB, and $a_{r,i} = 0$ if i^{th} user did not utilize the r^{th} of RB [6].

Figure 2 shows an example of an interference graph in D2D communication. Nodes represent communication, either cellular or D2D communication. The edges were interference between two connected nodes. Let U_1, U_3 , and U_4 were in the same RB, except U_2 . First cellular user (U_1) will be affected by interference from the D2D users (U_3 and U_4), as well as U_3 will be affected by interference from U_1 . $I_{i,j,r}$ was the edge weight which indicated the set of interference value from i^{th} node to j^{th} node at r^{th} resource block. Since U_1 's and U_2 's receiver is the eNB, if they shared a RB the receiver could not differentiate the signals. Thus, there was no interference between cellular users.



(b)

Figure 2. An example of an interference graph in D2D communication.

Maximum spectral efficiency of i^{th} communication at r^{th} RB, either a CUE or a DUE pair, can be calculated by:

$$Se_{i,r} = \frac{B \times \log_2(1 + SINR_{i,r})}{B_s} \quad (1)$$

where B is the bandwidth of the RB and B_s is the system bandwidth in Hertz, $SINR$ is the signal to interference plus noise ratio. The $SINR_{i,r}$ can be calculated by [6] :

$$SINR_{i,r} = \frac{a_{i,r} \cdot p_{i,r} \cdot g_{i,i,r}}{\sigma^2 + I_i^r} \quad (2)$$

$$I_i^r = \sum_{j=1, j \neq i}^{V+K} a_{j,r} \cdot p_{j,r} \cdot g_{j,i,r} \quad (3)$$

$$Se_{network} = \sum_{i=1}^{V+K} \sum_{r=1}^Q \log_2(1 + SINR_{i,r}) \quad (4)$$

$$\text{subject to} \begin{cases} C_1: \sum_{i=1}^V a_{i,r} \leq 1, & i \in M, \forall_r \in R \\ C_2: p_i < p_{dmax}, & i \in D \\ C_3: p_i < p_{cmax}, & i \in M \\ C_4: SINR_{i,r} \geq SINR_{min}, & i \in M, \forall_r \in R \end{cases} \quad (5)$$

In equation 2, $p_{i,r}$ is the transmission power of the i^{th} node at r^{th} RB, $g_{i,j,r}$ is the channel gain between the transmitter of the i^{th} node and receiver of the j^{th} node at r^{th} RB and σ^2 is the noise power. I_i^r is the total interference from other nodes to i^{th} node which can be written on equation 3. Equation 4 defines the network's spectral efficiency. Thus, the maximization problem defines in equation 5 [6]. C_1 make sure there will be at least 1 user on each RB. C_2 and C_3 limit the transmission power of each user. p_{dmax} and p_{cmax} are maximum transmission power of D2D users and cellular users, respectively. C_4 shows that $SINR_{min}$ is a minimum SINR for cellular users. The other simulation parameters can be seen on Table 1.

Table 1. Channel classification in mobile communication [6-8]

System Characteristics	Channel type
Cell radius	500 m
Channel model	COST-231 Hata model
Fading	Rayleigh
Shadowing	Lognormal shadowing; $\mu = 0$; $\sigma = 6$
Carrier frequency	1.8 GHz
Bandwidth of RBs	180 KHz
Maximum D2D pairs distance	10 m
Maximum power of DUE	23 dBm
Maximum power of CUE	23 dBm
Minimum power of DUE	-40 dBm
Noise power density	-174 dBm/Hz

4.1 Genetic Algorithm

The genetic algorithm consists of 3 main phases to form a new generation based on the current generation: proportional selection, crossover, and mutation. The algorithm started with a set of candidate solutions (chromosomes), which use dedicated power and RBs to communicate. Chromosomes should satisfy the constraint of the optimization problem in equation (5). The set of chromosomes was called a generation, that could be changed iteratively. Parents would be selected among the current generation to produce children (offspring) to form next generation. Parents will be selected based on the fitness of each chromosome, in terms of the spectral efficiency. After several iterations, the generation will achieve optimum value [6].

Algorithm 1 Genetic Algorithm for Fitness Calculation and Power Assignment

```

1 : procedure Fit_Calc (resourceVector, powerVector)
2 :   Output fitness_value, powerVector
3 :   Initialization: sum_vector =  $\emptyset$ ; cellVector =  $\emptyset$ ;
4 :   for RB_member each row in resourceVector
5 :     if there is any D2D pair in RB_member
6 :       Select optd randomly among D2D pairs of RB_member
7 :       for All possible values popt for power of optd
8 :         temp  $\leftarrow$  powerVector
9 :         Update temp with popt for optd
10 :        cell  $\leftarrow$  cellular user in RB_member
11 :        p(cell)  $\leftarrow$  required power for SINRmin
12 :        Add p(cell) to cellVector
13 :        if p(cell)  $\leq$  pc_max
14 :          Update temp with p(cell)
15 :          rate  $\leftarrow$  sumrate of RB_member with respect to temp
16 :        else
17 :          rate  $\leftarrow$  -infinity
18 :        end if
19 :        Add rate to sumVector
20 :        powerVector  $\leftarrow$  temp
21 :      end for
22 :      sumrate(RB_member)  $\leftarrow$  max(sumVector)
23 :      if All values of sumVector are -Inf
24 :        Update powerVector with pc_max and pD2D_max
25 :      else
26 :        p(cell)  $\leftarrow$  maximum value of cellVector in that RB
27 :        popt  $\leftarrow$  power of optd when the value of sum_vector was maximum
28 :        Update powerVector with p(cell) and popt
29 :      end if
30 :    else
31 :      p(cell)  $\leftarrow$  pc_max
32 :      Update powerVector with p(cell)
33 :      sumrate(RB_member)  $\leftarrow$  sumrate of RB_member with respect to powerVector
34 :    end if
35 :  end for
36 : fitness_value  $\leftarrow$  sum(sumrate)

```

Figure 2. Genetic algorithm for fitness and power assignment

4.1.1 Fitness Function and Power Assignment Calculation

The fitness function calculates the value of a chromosome. Since the purposes of the algorithm was to maximize the spectral efficiency of the network, so the fitness value of a chromosome equals to spectral efficiency of network according to equation (4). On each iteration, power of the users will be optimized. On each iteration, a random D2D communication (if any unallocated DUE) will be selected by each RB. The SINR of CUE on that corresponding RB will be set to $SINR_{min}$. Then, the power of i^{th} cellular user at r^{th} RB was calculated based on:

$$p_{i,r} = SINR_{min} \times \frac{\sigma^2 + \sum_{j=1, j \neq i}^{V+K} a_{j,r} \cdot p_{j,r} \cdot g_{j,i,r}}{a_{i,r} g_{i,i,r}} \quad (6)$$

However, if an RB was used by only one CUE, then the transmission power of that CUE will be maximum. The pseudocode for fitness calculation and power assignment can be seen on Figure 2. Chromosome (in this case is resource allocation matrix and transmission power vector), is being used as the inputs of the algorithm.

4.1.2 Proportional Selection

The proportional selection process selects chromosomes to become parents for next generation, which have the best fitness value in terms of spectral efficiency. If fitness value of the chromosome x denoted by $f_{chromosome}(x)$. The sets of $fit(t)$ which is the set of fitness values at t^{th} generation is denoted by [6]:

$$fit(t) = \{f_{chromosome}(x_1), f_{chromosome}(x_2), \dots, f_{chromosome}(x_{N_t})\} \quad (7)$$

where x_i and N_t are the i^{th} chromosome and the total number of chromosomes in t^{th} generation, respectively.

4.1.3 Crossover Process

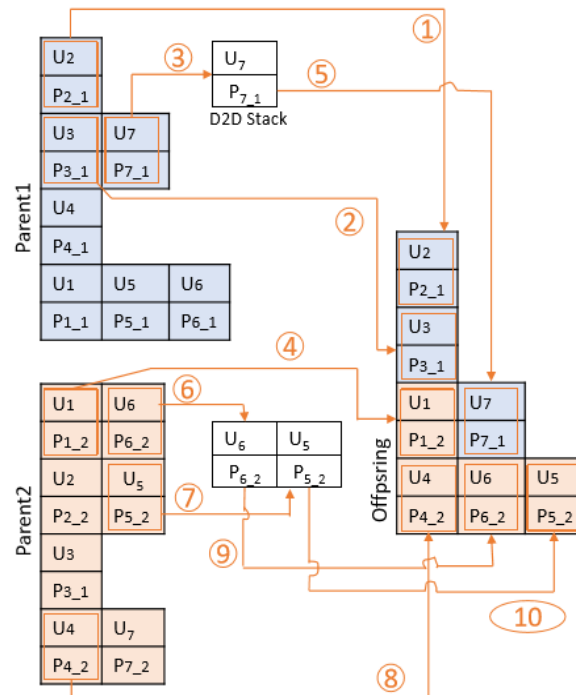
The crossover process will combine two “parents” and produces next generation’s offspring. On the initial stages, a random length of offspring is given to a RB. The length is equal to the total possible communications that can be supported in that RB. Next step is selecting a random number in range $[1, V+K]$ to be the crossover point (CV), which define the number of communication that have been taken from the parents. For example, $CP=3$ mean there are 3 communications (including the power transmit on that communications) are taken from parent1, and the rest of the length will be taken from parent 2. All detailed algorithm can be seen on Figure 3, the illustration can be seen on Figure 4. This process will create a local optimum solution, which will be the reference for the mutation process.

Algorithm 2 Genetic Algorithm for Crossover

```

1 : procedure Crossover (Parent1, Parent2)
2 :   Output offspring
3 :   Initialization: p=1, i=1
4 :   select Len = {len(1), len(2), ..., len(Q)} randomly in a way that
      : len(1)+len(2)+...+len(Q)=V+K
5 :   crossover_point ← random number 1 to V+K
6 :   if p ≤ crossover_point
7 :     temp(i) ← pth communication of the Parent1
8 :     i ← i+1
9 :   else
10 :    temp(i) ← (p-crossover_point)th communication of the Parent2
11 :    i ← i+1
12 :    while temp is already taken from Parent1
13 :      p ← p+1
14 :      temp(i) ← (p-crossover_point)th communication of the Parent2
15 :      i ← i+1
16 :    end while
17 :    i=1
18 :    for rb=1 to Q
19 :      for pos=1 to len(rb)
20 :        if pos is for a cellular user and temp(i) is a cellular user
21 :          if Cell_Stack is not empty
22 :            offspring(rb,pos) ← Cell_Stack(1)
23 :            Delete the first element in Cell_Stack
24 :          else
25 :            offspring(rb,pos) ← temp(i)
26 :            i ← i+1
27 :          end
28 :        else if pos is for a cellular user and temp(i) is a D2D pair
29 :          if Cell_Stack is not empty
30 :            offspring(rb,pos) ← Cell_Stack(1)
31 :            Delete the first element in Cell_Stack
32 :          else
33 :            while temp(i) is a D2D pair
34 :              Add temp to the D2D_stack
35 :              i ← i+1
36 :            end
37 :            offspring(rb,pos) ← temp(i)
38 :            i ← i+1
39 :          end
40 :        else if pos is for a D2D pair and temp(i) is a D2D pair
41 :          if D2D_Stack is not empty
42 :            offspring(rb,pos) ← D2D_Stack(1)
43 :            Delete the first element in D2D_Stack
44 :          else
45 :            offspring(rb,pos) ← temp(i)
46 :            i ← i+1
47 :          end
48 :        else if pos is for a D2D pair and temp(i) is a cellular user
49 :          if D2D_Stack is not empty
50 :            offspring(rb,pos) ← D2D_Stack(1)
51 :            Delete the first element in D2D_Stack
52 :          else
53 :            while temp(i) is a cellular user
54 :              Add temp(i) to the Cellular_stack
55 :              i ← i+1
56 :            end
57 :            offspring(rb,pos) ← temp(i)
58 :            i ← i+1
59 :          end
60 :        end if
61 :      end for
62 :    end for

```

Figure 3. Algorithm 2 genetic algorithm for crossover**Figure 4.** Crossover operation with CP is 3

Algorithm 3 Genetic Algorithm for Select and Reproduce

```

1 : procedure Sel_Rep (gen, fit)
2 :   Output new_gen
3 :   max_fit ← max(fit)
4 :   for chr = 1 to Nt
5 :     p1,p2 ← choose two parents based on the highest match value
6 :     new_in ← Crossover(p1,p2)
7 :     mut ← random number 0 to 1
8 :     if mut < 0.2
9 :       mut_gen ← Choose two cellular users and swap them
10 :      Add mut_gen to new_gen
11 :     else
12 :       Add new_in to new_gen
13 :     end if
14 :   end for

```

Figure 5. Genetic algorithm to select and reproduce**4.1.4 Mutation**

The next process is mutation. Mutation is a random change on the chromosome with low probability to escape from local optimum solution [9]. This random change is applied by swapping two random DUEs on corresponding RB. To prevent random search, the probability of mutation must be low. Combination of three main process proportional selection, crossover operation and mutation were called select and reproduce, and can be seen on Figure 5.

Algorithm 4 Genetic Algorithm for Optimization

```

1 : procedure Optimization (rep)
2 :   Output best_individual
3 :   Initialization ;
4 :   (initial_generation) ← random generation
5 :   i ← 1
6 :   do:
7 :     fit = ∅
8 :     for each chr in next_gen:
9 :       (fit_chr, Power) ← Fit_Calc(chr.resourceVector,chr.powerVector)
10 :      chr.powerVector ← Power
11 :      Add fit_chr to fit
12 :    end for
13 :    nextgen ← Sel_Rep(fit,next_gen)
14 :    i ← i+1
15 :  while i < rep
16 :  best_individual ← max(fit)

```

Figure 6. Genetic algorithm for optimization

4.1.5 Optimization using genetic algorithm

The steps of this process are explained as follows. At first, N_t random generation is generated as the first generation. The fitness value for first generation is also calculated with the maximum power of each user. Then, select and reproduce process will be executed to select the parents and produce offspring. Next the mutation process will take place on the offspring. This process will be repeated several times to calculate the best allocation in the system. The pseudocode of this process can be seen on Figure 6.

4.2 Greedy algorithm

The Greedy Algorithm is an algorithm that considers the data rate received by cellular users and D2D pairs. The greedy algorithm allocation process begun with, the first D2D pair chose the RB that belongs to the CU, where there was the highest total data rate on an RB. After the D2D pair selected one of the CUE devices, the other D2D pair couldn't select the RB that belongs to that CUE. This process was repeated until all D2D pairs had an RB. Figure 7 shows the pseudocode of the greedy algorithm.

Algorithm 5 Greedy Algorithm

```

1 : procedure Greedy (rate_total)
2 :   Output resourceVector_Greedy, sumrate_Greedy
3 :   for rb=1 to Q
4 :     do
5 :       max_rate ← a d2d communication choose an RB with
           the highest rate_total that does not have another d2d occupied
6 :       sumrate(rb) ← max_rate
7 :       while each D2D pair hasn't chosen RB
8 :     end for
9 :   sumrate_Greedy ← sum(sumrate)

```

Figure 7. Greedy algorithm

5. SIMULATION RESULTS AND ANALYSIS

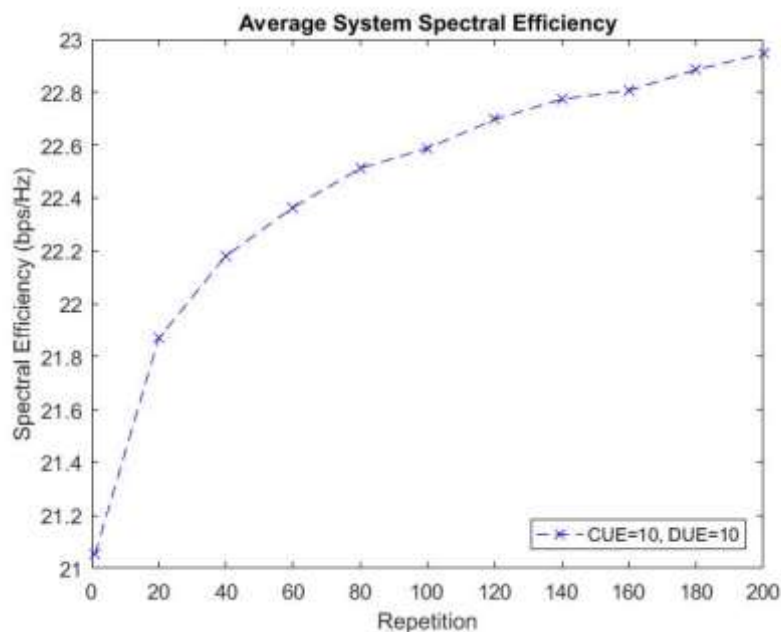
In this section, the performances of the Genetic algorithms are analyzed and compared with greedy algorithm. Several parameters that being observed are system sum-rate, system spectral efficiency, total power usage, system energy efficiency, and network interference level.

5.1 Varied Number of Repetition

At first, the genetic algorithm will be simulated independently with varies number of repetition. This simulation will observe the system's spectral efficiency in accordance with the original purpose. On this scenario, there are 10 CUEs and 10 DUEs on the cell. The results can be seen on Figure 8 and Table 2.

Table 2. Spectral efficiency of genetic algorithm

No.	Repetition	Spectral Efficiency (bps/Hz)
1	1	20.9845
2	20	21.7975
3	40	22.1319
4	60	22.3495
5	80	22.5275
6	100	22.5938
7	120	22.6758
8	140	22.7462
9	160	22.8128
10	180	22.8464
11	200	22.8940

**Figure 8.** System spectral efficiency for different number of repetitions

From the results can be seen that, with increased number of repetitions, the spectral efficiency of the system increased too. But with more repetitions, the increment of the spectral efficiency is decline. This means the algorithm have found the solution for the optimization problem. The select, reproduce, and mutation that happened on the algorithm will not have much effect when the solution is found.

For the next simulation, there will be 3 different algorithm that being observed. The greedy algorithm, genetic algorithm with only one repetition, and genetics algorithm with 100 times repetition. This scenario was made to give a better comparison among the algorithm. On the next scenario, there will be 10 CUEs on the cell with varies number of DUEs.

5.1 Sumrate

High system sum-rate will give a high data transmission speed so users can get more data. Simulation results on average system sumrate can be seen on Figure 9 and Table 3. The system sumrate will increase in line with the increasing number of DUE. The greedy algorithm is the best in this parameter. Greedy algorithm can achieve the average value of 45.36 Mbps. This happens because on greedy algorithm there are no power control scheme, so all user will get maximum power to communicate.

The genetic shows that the repetition really improve the algorithms performances. With 1 repetition, genetic algorithm only achieves 36.73 Mbps, while with 100 repetition this average number steps up to 39.97 Mbps. However, on 10 DUEs, the sumrate of 100 repetition genetics algorithm have best sumrate among the algorithm.

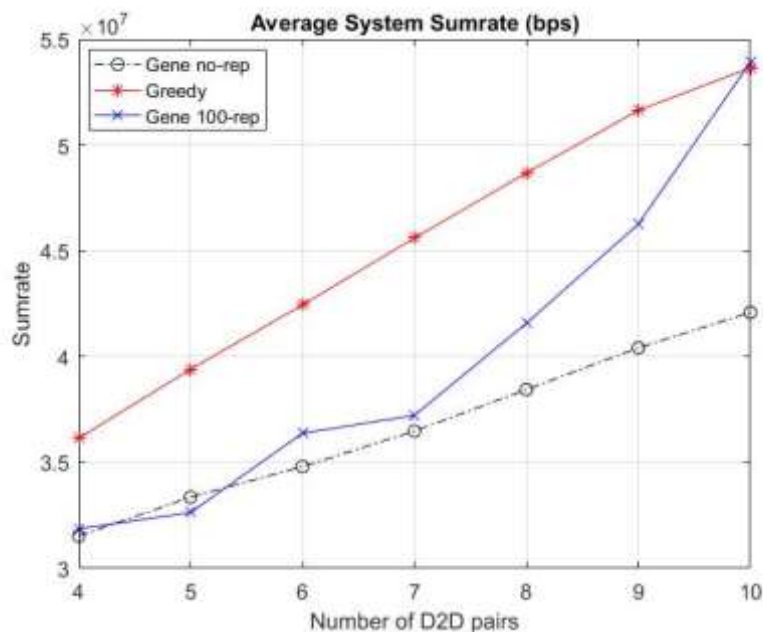


Figure 9. Sumrate of the system with varies number of DUEs

Table 3. Sumrate of the system

Number of DUE pairs	Sum-rate (x 10 ⁷ bps)		
	Genetic algorithm 1x repetition	Greedy algorithm	Genetic algorithm 100 repetition
4	3.1522	3.6141	3.1864
5	3.3358	3.9402	3.2635
6	3.4806	4.2445	3.6381
7	3.6482	4.5608	3.7225
8	3.8445	4.8678	4.1578
9	4.0415	5.1634	4.6266
10	4.2088	5.3647	5.3907
Mean	3.6731	4.5365	3.9979

5.2 Spectral Efficiency

Simulation results on average spectral efficiency of each algorithm can be seen on Figure 10 and Table 4. Same as the sumrate, spectral efficiency increases in line with the increasing number of DUE. The greedy algorithm is the best in this parameter. Greedy algorithm can achieve the average value of 22.68 bps/Hz. This happens because on greedy algorithm there are no power control scheme, so all user will get maximum power to communicate. This result also caused by the greedy algorithm's constraint that limit the user on each RB with only 1 CUE and 1 DUE pair, so the SINR value can be maintained.

The genetic shows that the repetition really improve the algorithms performances. With 1 repetition, genetic algorithm only achieves 18.36 bps/Hz, while with 100 repetition this average number steps up to 19.78 bps/Hz. But overall, genetic algorithm with 100 repetitions will have more complexity because of the repetition itself, while the other algorithm only uses 1 loop of the process.

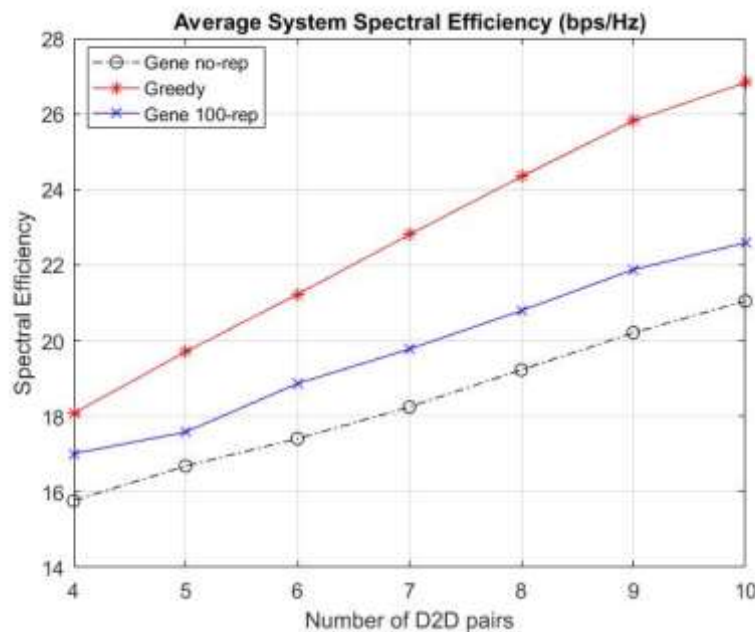


Figure 10. Spectral efficiency of the system with varies number of DUEs

Table 4. Spectral efficiency of the system

Number of DUE pairs	Spectral efficiency (bps/Hz)		
	Genetic algorithm 1 repetition	Greedy algorithm	Genetic algorithm 100 repetition
4	15.7609	18.0707	17.0057
5	16.6788	19.7012	17.5785
6	17.4031	21.2227	18.8623
7	18.241	22.8039	19.7728
8	19.2223	24.3389	20.7919
9	20.2076	25.8169	21.8784
10	21.0441	26.8233	22.5877
Mean	18.3654	22.6825	19.7825

5.3 Total Power Usage

Total power usage in the system is calculated by summing all power transmitted in the system, including CUEs and DUEs. The simulation results can be seen on Figure 11 and Table 5. From Figure 11, can be concluded that the genetics algorithm with 100 repetitions have the lowest power usage among all algorithm.

Overall, all algorithm power usage increase in line with increasing number of DUEs. But in genetic algorithm, the power usage is more controlled. This happens because in the initial steps in genetic algorithm, there are power control scheme. The power control scheme limits the power transmit on all users. On 1 repetition, average total power usage is 2.17 W. The repetition will decrease this number again, because in each repetition this usage power will be maximized to achieve better spectral efficiency on each RB. Meanwhile, the greedy algorithm, which use fixed power, the increasing number of users will increased the total power usage linearly.

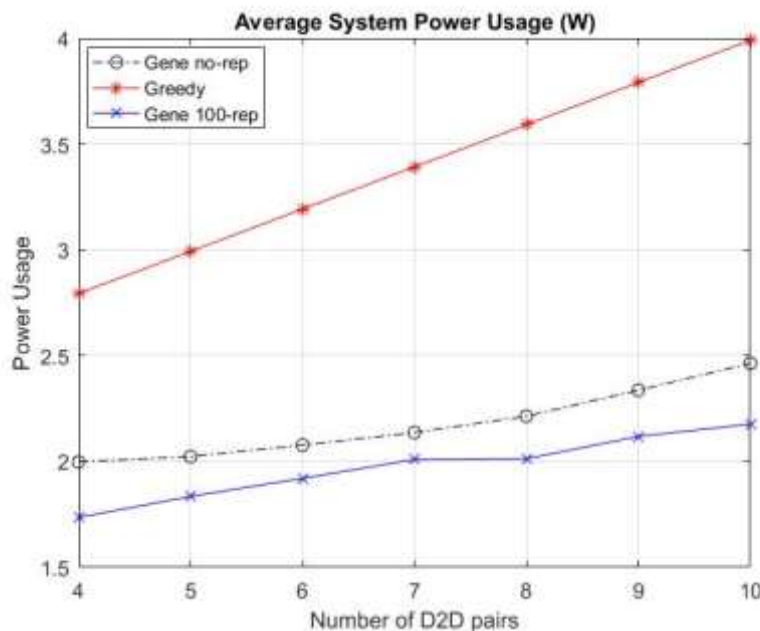


Figure 11. Total power usage in the system with varies number of DUEs

Table 5. Total power usage

Number of DUE pairs	Total power system (W)		
	Genetic algorithm 1 repetition	Greedy algorithm	Genetic algorithm 100 repetition
4	1.9975	2.7934	1.7355
5	2.0234	2.9929	1.8347
6	2.0773	3.1924	1.9200
7	2.1364	3.3919	2.0092
8	2.2137	3.5915	2.0127
9	2.3366	3.791	2.1180
10	2.4636	3.9905	2.1769
Mean	2.1783	3.3919	1.9724

5.4 Energy Efficiency

Energy efficiency means how many bit per second can be transferred in 1 Watt of power. The unit is bps/W (bit per second per Watt). The results of the simulation can be seen on Figure 12 and Table 6. In this parameters, genetic algorithm produce the best value. The 1 repetition genetic algorithm, the average energy efficiency is 17.07 Mbps/W. This value jumps with 100 repetitions which achieve 20.27 Mbps/W. Meanwhile greedy algorithm only achieve 13.35 Mbps/W in average.

This condition also happens caused by the power control scheme that included in the genetic algorithm. This scheme can minimize the power transmit on each user but still maximize the SINR on each RB. Greedy algorithm prove that with maximum transmitted power, the spectral efficiency can be maximized, but can be a waste in energy matters.

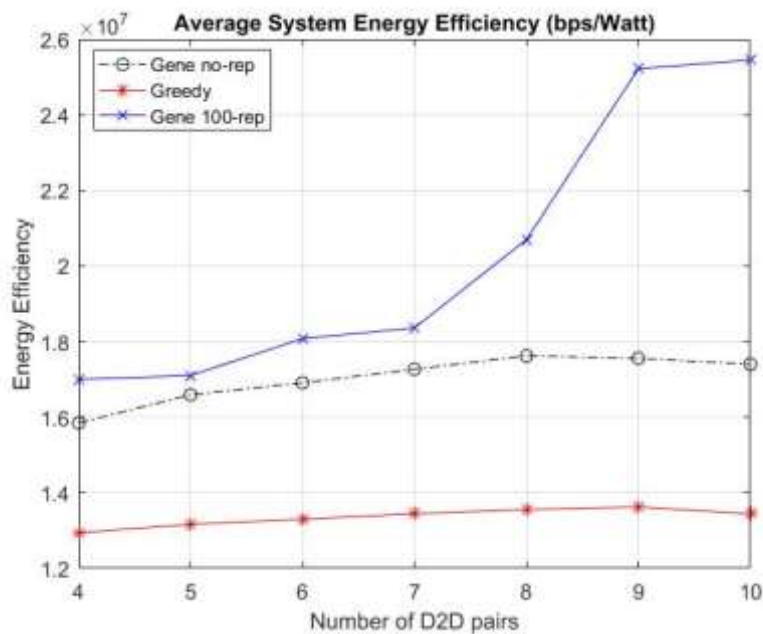


Figure 12. Energy efficiency of the system with varies number of DUEs

Table 6. Energy efficiency of the system

Number of DUE pairs	Energy efficiency (x 10 ⁷ bps/W)		
	Genetic algorithm 1 repetition	Greedy algorithm	Genetic algorithm 100 repetition
4	1.5846	1.2938	1.6997
5	1.6589	1.3165	1.7101
6	1.6908	1.3296	1.8076
7	1.7267	1.3446	1.8360
8	1.7621	1.3554	2.0694
9	1.7555	1.362	2.5217
10	1.7398	1.3443	2.5452
Mean	1.7026	1.3352	2.0271

5.5 Network Interference

The total interference that happens in the cell on each algorithm can be seen on Figure 13 and Table 7. The genetic algorithm also shows the best result in this parameter. Genetic algorithm can achieve -59.24 dBm in average of interference level on 1 repetition, and can achieve a lower average interference which is -98.40 dBm with 100 repetition. Meanwhile the interference level on greedy algorithm only -53.86 dBm in average.

This result means the genetic algorithm can control and maintained the interference level in the cell. By decreasing and limiting each transmitted power in each user, the interference level on the system can stay at the lower level.

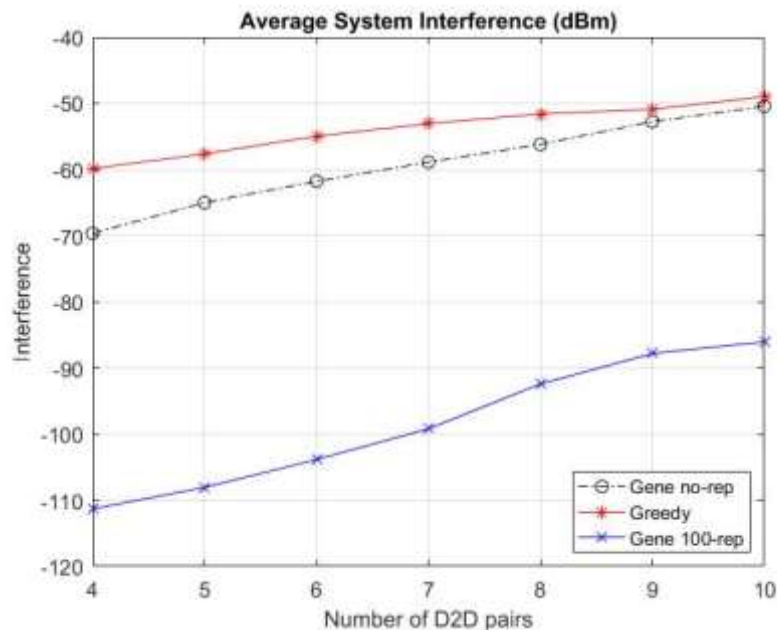


Figure 13. Network Interference with varies number of DUEs

Table 7. Network Interference

Number of DUE pairs	Network Interference (dBm)		
	Genetic algorithm 1 repetition	Greedy algorithm	Genetic algorithm 100 repetition
4	-69.6518	-59.8589	-111.3231
5	-65.0128	-57.6444	-108.0780
6	-61.7879	-54.9705	-103.8633
7	-58.8895	-53.0778	-99.2212
8	-56.1939	-51.5917	-92.4305
9	-52.7546	-50.9153	-87.8322
10	-50.445	-48.9822	-86.0584
Mean	-59.2479	-53.863	-98.4010

Table 8. Overall comparison among algorithm

Parameter	Algorithm's mean value		
	Greedy	Genetic 1 Repetition	Genetic 100 Repetitions
Sumrate	45.36 Mbps	36.73 Mbps	39.97 Mbps
Spectral Efficiency	18.07 bps/Hz	15.76 bps/Hz	17.00 bps/Hz
Energy Efficiency	13.35 Mbps/W	17.02 Mbps/W	20.27 Mbps/W
Power Usage	3.39 Watt	2.17 Watt	1.97 Watt
Network Interference	-53.86 dBm	-59.24 dBm	-98.40 dBm

5.6 Simulation Analysis

Can be seen on the simulation results, Genetic algorithm has the excellence in energy efficiency, while greedy algorithm has a good spectral efficiency. The summary of the simulation results can be seen on Table 8, with the best value on each parameter is coloured grey.

The Greedy has more in sumrate and spectral efficiency because it restricts each RB to only 2 users in one time maximum, which are 1 CUE and 1 DUE. The spectral efficiency can also be maximized by allocating maximum power on each user. But this scheme caused a depression in the energy efficiency and increasing interference among users in the system. This algorithm can cause a power wastage in the system.

Meanwhile in genetic algorithm, the power control and the repetition process maximized the energy efficiency and lowering the power usage and the interference level in the cell. By allocating each DUE according to the fitness of each RB, and allocating each user's power according to the condition of each RB, genetic algorithm can avoid power wastage in the system. In other hand, the spectral efficiency cannot be maximized. By increasing the repetition on the algorithm also makes the computational process on genetic algorithm requires more time, so the system complexity also increases. These results mean there are trade-offs among the algorithms.

6. CONCLUSION

In this research, the performances of genetic algorithm on D2D underlying communication are analyzed, and then compared with the conventional greedy algorithm. The greedy algorithm allocates only based on the best SINR for each DUE, and allocates maximum fixed power on each user, without considering the other parameters. The results show that the greedy algorithm achieves higher value in sumrate and spectral efficiency, averaging 45.36 Mbps and 18.07 bps/Hz respectively. The genetic algorithm only achieves an average sumrate at 39.97 Mbps and average spectral efficiency at 17.00 bps/Hz.

In the other hand, by allocating user and transmit power on each user with the purpose of maximizing the fitness value (in this case is spectral efficiency), genetic algorithm cannot achieve higher spectral efficiency compared to greedy algorithm. Although, genetic algorithm achieves better value in energy efficiency with average value in 20.27 Mbps/Watt. The

system's power usage also the lowest at 1.97 Watt, when greedy algorithm uses 3.39 Watt in average. The genetic algorithm also can maintain the system's interference level on -98.40 dBm.

REFERENCES

- [1] R. D. Ainul, H. Mahmudah, A. Wijayanti, **Performance Analysis of Scheduling Schemes for Femto to Macro Interference Coordination in LTE-Femtocell Deployment Scenario**, EMITTER International Journal of Engineering Technology, vol. 4, no. 11, pp 65-90, 2016
- [2] Afif Osseiranetal, **Scenarios for 5G mobile and wireless communications: The vision of the METIS project**, IEEE Commun. Mag., vol. 52, no. 5, pp. 26–35, 2014.
- [3] Li Wang, Huan Tang, **Device-to-Device Communications in Cellular Networks**, *Springer International Publishing*, 2016.
- [4] Y. Priatama, **Radio Resource Management and Interference Mitigation for Device to Device Communication Underlying 5G Network**, Undergraduated Thesis, Telkom University (Bandung), 2019.
- [5] Guanding Yu, Lukai Xu, Daquan Feng, Rui Yin, Geoffrey Ye Li, Yuhuan Jiang, **Joint mode selection and resource allocation for device-to-device communications**, IEEE Trans. Commun., vol. 62, no. 11, pp. 38143824, 2014.
- [6] Hengameh Takshi, Gulustan Dogan, Huseyin Arslan, **Joint Optimization of Device to Device Resource and Power Allocation Based on Genetic Algorithm**, IEEE Access, vol. 6, pp. 21173–83, 2018.
- [7] V. S. W. Prabowo, B. Pamukti, A. Fahmi, N. M. Adriansyah and N. Andini, **Joint-Greedy Allocation Algorithm on D2D Communication Underlying Networks**, 2019 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), BALI, Indonesia, 2019, pp. 48-52.
- [8] H. Ding, S. Ma, and C. Xing, **'Feasible D2D communication distance in D2D-enabled cellular networks,'** in Proc. IEEE Int. Conf. Commun. Syst. (ICCS), Nov. 2014, pp. 1–5.
- [9] K. Deb, **Multi-Objective Optimization Using Evolutionary Algorithms**, vol. 16. Hoboken, NJ, USA: Wiley, 2001.